Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs
Generate OACS

Search Results - Record(s) 1 through 28 of 28 returned.

□ 1. Document ID: US 20030115423 A1

Using default format because multiple data bases are involved.

L2: Entry 1 of 28 File: PGPB Jun 19, 2003

PGPUB-DOCUMENT-NUMBER: 20030115423
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20030115423 A1

TITLE: Cache states for multiprocessor cache coherency protocols

PUBLICATION-DATE: June 19, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Chang, Stephen S.	Glendora	CA	US	

US-CL-CURRENT: 711/145; 711/144, 711/146

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

□ 2. Document ID: US 20030046495 A1

L2: Entry 2 of 28 File: PGPB Mar 6, 2003

DOCUMENT-IDENTIFIER: US 20030046495 A1
TITLE: Streamlined cache coherency protocol system and method for a multiple processor single chip device

Detail Description Paragraph:
[0027] In the present invention cache coherency protocol architecture, each cache line has a state. There are three primary cache line states, Modified (M), Shared (S), and Invalid (I). An attempted access or transaction (e.g., read or write) to a line in a cache can have different consequences depending on whether it is an internal access by the processor core, or an external access by another processor core on the system bus (e.g., system bus 110) or another memory (e.g., eDRAM 130). A cache line in an invalid state is not available for access in the particular cache in which it is in the invalid state. An internal access to a line misses the cache and will cause the processor core to fetch the line information from the system bus (e.g., fetch information from eDRAM 130 or from another cache in another processor core via system bus 110). In a shared cache line state the line comprises the same value as in memory, and can have a shared state in other caches.

Internally reading a shared state cache line causes no system bus activity. Attempting to internally write the cache line causes a cache line in other caches to enter an invalidate state line before the internal write is permitted to proceed. In the modified (M) cache line state the line includes a more recent value than memory, and is invalid in other caches. Internally reading or writing the modified state cache line causes no system bus activity.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

□ 3. Document ID: US 20030037096 A1

L2: Entry 3 of 28 File: PGPB Feb 20, 2003

DOCUMENT-IDENTIFIER: US 20030037096 A1
TITLE: METHOD AND APPARATUS FOR THE MANAGEMENT OF QUEUE POINTERS BY MULTIPLE PROCESSORS IN A DIGITAL COMMUNICATIONS NETWORK

Detail Description Paragraph:
[0038] Wherein caches 41 and 46 are shown as being internal to the respective processors, caches 41 and 46 may be external to the processors. Furthermore, the head and tail pointers may alternatively be stored by internal registers of the queuing and servicing processors, or by external registers. Because the head and tail pointers are independent from one another, many of the advantages of the presently described methods may be achieved by storing the head and tail pointers in non-shared memory locations such that atomic memory accesses are not required.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

□ 4. Document ID: US 20020120709 A1

L2: Entry 4 of 28 File: PGPB Aug 29, 2002

DOCUMENT-IDENTIFIER: US 20020120709 A1
TITLE: Shared-memory controller for use in a multimedia processor system

Abstract Paragraph:
With a shared common memory, internal requests of a multimedia processor (70) is processed to generate a request (140) optimized in format. Each internal request is selected by the request controller according to a system priority. Each of external requests is independently subjected to an analyzing process for each format adapter (160), and is converted into a format compatible with a predetermined internal format of the multimedia processor. Thus, one or more external processors share a common memory to enable the external processors to access a bulk memory storage of a multimedia processor.

Summary of Invention Paragraph:
[0032] According to a second aspect of the present invention, a shared-memory controller uses an external request from an external processor associated with a multimedia processor and an internal request of the multimedia processor to enable

the external processor to perform memory access to a main memory. The shared-memory controller includes: internal-request control means for sequentially controlling in selection of the individual internal requests; formatter means for format-optimizing the selected internal request to generate an optimally-formatted request; and format-adapter means for format-adapting the external request to generate an adapted request suitable to a predetermined internal format of the shared-memory controller, The shared-memory controller further includes: prioritizing means receiving the format-optimized request and the format-adapted request and performing priority arbitration for memory access requests; and request/command converting means for converting the priority-arbitrated request into a memory-access command.

CLAIMS:

4. A shared-memory controller which uses an external request from an external processor associated with a multimedia processor and an internal request of said multimedia processor to enable the external processor to perform memory access to a main memory, comprising: internal-request control means for sequentially controlling in selection of the individual internal requests; formatter means for format-optimizing the selected internal request to generate an optimally-formatted request; format-adapter means for format-adapting the external request to generate an adapted request suitable to a predetermined internal format of the shared-memory controller; prioritizing means receiving the format-optimized request and the format-adapted request and performing priority arbitration for memory access requests; and request/command converting means for converting the priority-arbitrated request into a memory-access command.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

□ 5. Document ID: US 6694409 B2

L2: Entry 5 of 28 File: USPT Feb 17, 2004

DOCUMENT-IDENTIFIER: US 6694409 B2
TITLE: Cache states for multiprocessor cache coherency protocols

Detailed Description Text (6):
FIG. 1 is a diagram of a typical hardware and operating environment in conjunction with which embodiments of the invention may be implemented. In many computer systems, a processor is coupled to an internal or external cache, and to an external memory device over a high speed memory bus. The example computer system 100 shown in FIG. 1 comprises a plurality of processors 102. Each one of the processors is coupled to a cache memory 104. At least one bus 108 interconnects the shared main memory 106 with the plurality of cache memories 104 and the multiple processors 102. When data is read from the shared main memory 106 and used by one of the processors 102, the data and its main memory address are also stored in the processor's cache memory 104. Data from a given main memory location can reside simultaneously in main memory 106 and in one or more cache memories 104. Also, the data from a given main memory location found in one cache memory 104 may have a value different from that in the main memory 106 because one has been updated more recently.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw. De

□ 6. Document ID: US 6668275 B1

L2: Entry 6 of 28 File: USPT Dec 23, 2003

DOCUMENT-IDENTIFIER: US 6668275 B1
TITLE: System and method for multiprocessor management

Brief Summary Text (9):
The present invention provides a system for managing the operation and maintenance of complex multiprocessor systems. In accordance with one aspect of the present invention, a system is disclosed for managing a multiprocessor system includes a core processor in communication with at least one remote processor or process or thread or tasks running on at least one machine. A core processor interacts with the distributed processes through internal or external communication channels such as messaging queues, packets, block transfers or shared memory schemes. The core provides test, synchronization, logging, and management operations to the distributed processors. An agent engine such as a script interpreter, associated with the core, provides operation-related data and software from a combination of user interfaces and databases to the core during its interaction with the distributed processors. Target agents associated with the distributed processors receive commands from the core. Target agents can access a library module for an extended command set for processor management.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw. De

□ 7. Document ID: US 6654835 B1

L2: Entry 7 of 28 File: USPT Nov 25, 2003

DOCUMENT-IDENTIFIER: US 6654835 B1
TITLE: High bandwidth data transfer employing a multi-mode, shared line buffer

Detailed Description Text (22):
By way of further example, using the configuration of FIGS. 5 & 6, an external bus master may access any valid dedicated memory address of the integrated device. This again requires that the DMA controller and system interface be properly set up. A "memory to memory line mode" operation should be set up for the internal DMA controller to correctly communicate with the shared line buffer. Additionally, the bus interface should be configured to access the shared line buffer as an SRAM in burst mode. The block sizes supported for this mode must be a multiple of 16 bytes and can be as large as the DMA controllers allow. By way of example, the following sequence of actions may be performed for a transfer in this mode: The DMA controller is set up to use the shared line buffer and transfer the appropriate size data block. This set up may be done by application code running on either the internal microprocessor or the external processor. The external processor sets up its own DMA controller to access the integrated device and transfer the appropriate size data block. The line buffer address should be used as the target address for the DMA transfer. The line buffer is set up by indicating the mode (DMA v. IDE) and

direction (read/write) of the transfer. The operation is then initiated by setting the line buffer active bit in the arbiter register. Again, this set up may be done by either the internal processor or the external processor. The line buffer will then drive the transfer by alternately transferring, for example, 16 byte blocks between the external processor and line buffer, and the line buffer and the DMA controller of the integrated device. Completion of the operation is signaled by either or both of the DMA controllers completing their respective transfers.

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw De

□ 8. Document ID: US 6606677 B1

L2: Entry 8 of 28 File: USPT Aug 12, 2003

DOCUMENT-IDENTIFIER: US 6606677 B1
** See image for Certificate of Correction **
TITLE: High speed interrupt controller

Detailed Description Text (6):
FIG. 1 illustrates, in a form of a block diagram, an exemplary embodiment of the present invention of a data communication system. The system incorporates a host adapter circuit 5, used between a host processor 22, which is the main system processor with an external central shared memory 20, and an internal processor system 14, connected to at least one external peripheral device interface 350 of FIG. 3. FIG. 3 illustrates a block diagram of the internal processor system 14 of FIG. 1, utilizing a high speed interrupt controller 318 of the present invention. The internal processor system 14 consists of an internal processor 314, an associated instruction and data memory 313, and the high speed interrupt controller 318, all connected to an internal bus 16, via a memory controller/bus bridge 315. It also consists of at least one external peripheral device interface 350, also connected to the internal bus 16, via a bus adapter2317. The internal processor system 14 is controlled by the processor 314. The internal processor 314 is connected to the data memory 313 via a processor bus 321. This architecture may be used as an adapter or a bridge between the host processor 22 and the peripheral device interface 350, and includes logic, according to the present invention, which allows data transfers between the peripheral devices, not shown, attached to the peripheral device interface(s) 350, and the central shared memory 20. The high speed interrupt controller 318 is used for assisting the processor 314, which has a limited number of interrupt input lines, to quickly discriminate between interrupts from different sources.

Detailed Description Text (39):
Each bus master 202 can be instructed to request the shared bus 200 in one of two ways. In the first method, an initiator processor on an external bus 214 or internal bus 216 wishes to write/read a short message to/from a target component, such as the central shared memory or another processor, on another internal bus 216 or external bus 214, or from/to one of the memory arrays 218. In the second method, a DMA channel within the master 202 circuit is instructed by the initiator processor to perform a long burst transaction (either write or read) between its associated internal bus 216 or external bus 214 and one of the slaves 212, for example, to transfer data between the central shared memory and one of the memory arrays 218.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw. De

□ 9. Document ID: US 6539500 B1

L2: Entry 9 of 28 File: USPT Mar 25, 2003

DOCUMENT-IDENTIFIER: US 6539500 B1
TITLE: System and method for tracing

Detailed Description Text (9):
FIG. 2 illustrates another embodiment of the present invention where the trace buffer resides in shared memory 202 external to processors 100 and 103. If shared memory used to implement the trace buffer 108 was external to the processors there would exist more flexibility in setting the size of the trace buffer. When the shared memory used for the trace buffer is internal to the processors and the processors are on one chip it may be difficult to estimate the size of a trace buffer needed for all trace applications. The embodiment of the present invention illustrated in FIG. 2 gives more flexibility without loss of generality. In yet another embodiment of the present invention both internal and external shared memory connected with special bus 200 could be used for trace buffer 108.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw. De

□ 10. Document ID: US 6523060 B1

L2: Entry 10 of 28 File: USPT Feb 18, 2003

DOCUMENT-IDENTIFIER: US 6523060 B1
TITLE: Method and apparatus for the management of queue pointers by multiple processors in a digital communications network

Detailed Description Text (11):
Wherein caches 41 and 46 are shown as being internal to the respective processors, caches 41 and 46 may be external to the processors. Furthermore, the head and tail pointers may alternatively be stored by internal registers of the queuing and servicing processors, or by external registers. Because the head and tail pointers are independent from one another, many of the advantages of the presently described methods may be achieved by storing the head and tail pointers in non-shared memory locations such that atomic memory accesses are not required.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw. De

□ 11. Document ID: US 6519685 B1

L2: Entry 11 of 28 File: USPT Feb 11, 2003

DOCUMENT-IDENTIFIER: US 6519685 B1
**** See image for Certificate of Correction ****
TITLE: Cache states for multiprocessor cache coherency protocols

Detailed Description Text (6):
FIG. 1 is a diagram of a typical hardware and operating environment in conjunction with which embodiments of the invention may be implemented. In many computer systems, a processor is coupled to an internal or external cache, and to an external memory device over a high speed memory bus. The example computer system 100 shown in FIG. 1 comprises a plurality of processors 102. Each one of the processors is coupled to a cache memory 104. At least one bus 108 interconnects the shared main memory 106 with the plurality of cache memories 104 and the multiple processors 102. When data is read from the shared main memory 106 and used by one of the processors 102, the data and its main memory address are also stored in the processor's cache memory 104. Data from a given main memory location can reside simultaneously in main memory 106 and in one or more cache memories 104. Also, the data from a given main memory location found in one cache memory 104 may have a value different from that in the main memory 106 because one has been updated more recently.

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw De

□ 12. Document ID: US 6519666 B1
L2: Entry 12 of 28 File: USPT Feb 11, 2003

DOCUMENT-IDENTIFIER: US 6519666 B1
TITLE: Arbitration scheme for optimal performance

Detailed Description Text (33):
Each bus master 202 can be instructed to request the shared bus 200 in one of two ways. In the first method, an initiator processor on an external bus 214 or internal bus 216 wishes to write/read a short message to/from a target component, such as the central shared memory or another processor, on another internal bus 216 or external bus 214, or from/to one of the memory arrays 218. In the second method, a DMA channel within the master 202 circuit is instructed by the initiator processor to perform a long burst transaction (either write or read) between its associated internal bus 216 or external bus 214 and one of the slaves 212, for example, to transfer data between the central shared memory and one of the memory arrays 218.

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw De

□ 13. Document ID: US 6496890 B1
L2: Entry 13 of 28 File: USPT Dec 17, 2002

DOCUMENT-IDENTIFIER: US 6496890 B1
TITLE: Bus hang prevention and recovery for data communication systems employing a

shared bus interface with multiple bus masters

Detailed Description Text (39):
Each bus master 202 can be instructed to request the shared bus 200 in one of two ways. In the first method, an initiator processor on an external bus 214 or internal bus 216 wishes to write/read a short message to/from a target component, such as the central shared memory or another processor, on another internal bus 216 or external bus 214, or from/to one of the memory arrays 218. In the second method, a DMA channel within the master 202 circuit is instructed by the initiator processor to perform a long burst transaction (either write or read) between its associated internal bus 216 or external bus 214 and one of the slaves 212, for example, to transfer data between the central shared memory and one of the memory arrays 218.

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw De

□ 14. Document ID: US 6360301 B1

L2: Entry 14 of 28 File: USPT Mar 19, 2002

DOCUMENT-IDENTIFIER: US 6360301 B1
TITLE: Coherency protocol for computer cache

Detailed Description Text (2):
The invention is applicable to any multi-level cache system, but is particularly advantageous in large multiprocessor systems. FIG. 3 illustrates an example of a computer system in which the present invention is particularly applicable. In FIG. 3, a computer system has N processors, two of which are illustrated (300, 302). Each processor has three levels of internal caches (304, 306, 308 and 310, 312, 314) and a fourth external cache (316, 318). All processors and their associated cache hierarchies share a system bus 320 and a system memory 322. Bus 324 illustrates that multiple processors may share an external cache, such as cache 316. The invention deals with a cache coherency protocol that might be used for any lower level cache, for example, the two L3 caches (316 and 318) in FIG. 3. In a system in accordance with the invention, a lower level cache, for example cache 316, detects when a line is evicted from a higher level cache. If a line has been evicted from a higher level cache, then there is no need for a back-invalidate transaction when the line is evicted from the lower level cache. Accordingly, the lower level cache coherency protocol includes an additional state that indicates that a line is not cached at higher levels, and therefore does not require a back-invalidate transaction when evicted. Reduction of back-invalidate transactions improves performance.

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw De

□ 15. Document ID: US 6314500 B1

L2: Entry 15 of 28 File: USPT Nov 6, 2001

DOCUMENT-IDENTIFIER: US 6314500 B1
TITLE: Selective routing of data in a multi-level memory architecture based on source identification information

Detailed Description Text (2):
The illustrated implementations of the invention generally operate by selectively routing requested data to memory sources associated with a particular requester in response to source identification information supplied by the memory source that is sourcing the requested data. A requester may be a processor or processing unit, or any other logic circuitry that utilizes data stored in a shared memory system, e.g., input/output adapters and/or interfaces, memory controllers, cache controllers, etc. A memory source, in turn, can include practically any data storage device or subsystem in a shared memory system from which identification and/or state information may be maintained, including main storage and various levels of cache memories, irrespective of the level of such cache memories, whether such cache memories are internal or external relative to a processor or other requester, whether such cache memories are data-only memories or collective data/instruction memories, whether such cache memories are dedicated to a particular requester or shared among several requesters, etc. A memory source can also include other shared or dedicated memories, including virtual memory, e.g., as implemented with one or more direct access storage devices in a page-based memory system. A memory source may also include memories distributed in a cache-only memory architecture (COMA) or a non-uniform memory architecture (NUMA) system. Furthermore, a memory source can also include other buffers or registers that may serve as a source for data, including translation lookaside buffers, processor registers, processor buffers, etc.

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw De

□ 16. Document ID: US 6260090 B1

L2: Entry 16 of 28 File: USPT Jul 10, 2001

DOCUMENT-IDENTIFIER: US 6260090 B1
TITLE: Circuit arrangement and method incorporating data buffer with priority-based data storage

Detailed Description Text (3):
A memory requester as used herein may be considered to be a processor or processing unit, or any other logic circuitry that utilizes data stored in a shared memory system. A data source, in turn, may include practically any data storage device or subsystem in a shared memory system from which data may be retrieved, including main storage and various levels of cache memories, irrespective of the level of such cache memories, whether such cache memories are internal or external relative to a processor or other requester, whether such cache memories are data-only memories or collective data/instruction memories, whether such cache memories are dedicated to a particular requester or shared among several requesters, etc. A data source can also include other shared or dedicated memories, including virtual memory, e.g., as implemented with one or more direct access storage devices in a page-based memory system. A data source may also include memories distributed in a cache-only memory architecture (COMA) or a non-uniform memory architecture (NUMA) system. Furthermore, a data source can also include other buffers or registers that may serve as a source for data, including translation lookaside buffers, processor registers, processor buffers, etc. Further, a data source may include any

combination of the above components from which a common response to a memory request can be generated.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

□ 17. Document ID: US 6101255 A

L2: Entry 17 of 28 File: USPT Aug 8, 2000

DOCUMENT-IDENTIFIER: US 6101255 A
TITLE: Programmable cryptographic processing system and method

Detailed Description Text (4):
FIG. 1 illustrates a hardware block diagram of a programmable crypto processing system in accordance with a preferred embodiment of the present invention. Crypto processing system 10 has, in the preferred embodiment, two primary processing elements, Key management crypto engine (KMCE) 12 and programmable cryptographic processor (PCP) 17. PCP 17 comprises two processing engines, programmable cryptographic engine (PCE) 14 and configurable cryptographic engine (CCE) 16. The processing engines perform the execution of channel programs. System 10 also includes cryptographic controller (CC) 11 which performs the program management for the processing engines. System 10 also includes plane text interface processor (PTIP) 13 and cipher text interface processor (CTIP) 15 which provide external interfaces and signaling for system 10. The interface processors also provide a high performance secure flexible buffer between an external host and the internal processing systems of system 10. System 10 also includes shared memory 18 which serves as an elastic buffer between KMCE 12 and PCP 17. System 10 also includes programmable interface 21 coupled with FILL and CIK ports 34. Testing of system 10 may be performed using test interface 20 which includes on-chip emulation and JTAG ports 35.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

□ 18. Document ID: US 6081895 A

L2: Entry 18 of 28 File: USPT Jun 27, 2000

DOCUMENT-IDENTIFIER: US 6081895 A
TITLE: Method and system for managing data unit processing

Detailed Description Text (5):
controller (CC) 100 which performs the program management for the processing engines. System 10 also includes plain text interface processor (PTIP) 13 and cipher text interface processor (CTIP) 15 which provide external interfaces and signaling for system 10. The interface processors also provide a high performance secure flexible buffer between an external host and the internal processing systems of system 10. System 10 also includes shared memory 18 which serves as an elastic buffer between KMCE 12 and PCP 17. System 10 also includes programmable interface 21 coupled with FILL and CIK ports 34. Testing of system 10 may be performed using

test interface 20 which includes on-chip emulation and JTAG ports 35.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

□ 19. Document ID: US 6014729 A

L2: Entry 19 of 28 File: USPT Jan 11, 2000

DOCUMENT-IDENTIFIER: US 6014729 A
TITLE: Shared memory arbitration apparatus and method

Detailed Description Text (21):
The M-bus and the SH-bus are coupled by the arbitration circuitry of the present invention. This provides the processor with access to shared memory allowing it to load and store data and to transfer data between the 1553 interface and the I/O devices. The arbitration circuitry is comprised of two major components: the arbitration logic, 190; and the buffer, 192. Because the ability to disable the processor and use an external processor is present in the preferred embodiment, the arbitration scheme must be able to work with both an internal and an external processor. This capability is enabled by providing the connection between the buses, instead of to the processor, and by making the signals passed between the processor and the arbitration logic available on external pins.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

□ 20. Document ID: US 5961626 A

L2: Entry 20 of 28 File: USPT Oct 5, 1999

DOCUMENT-IDENTIFIER: US 5961626 A
TITLE: Method and processing interface for transferring data between host systems and a packetized processing system

Detailed Description Text (4):
FIG. 1 illustrates a hardware block diagram of a programmable crypto processing system in accordance with a preferred embodiment of the present invention. Crypto processing system 10 has, in the preferred embodiment, two primary processing elements, Key management crypto engine (KMCE) 12 and programmable cryptographic processor (PCP) 17. PCP 17 comprises two processing engines, programmable cryptographic engine (PCE) 14 and configurable cryptographic engine (CCE) 16. The processing engines perform the execution of channel programs. System 10 also includes cryptographic controller (CC) 100 which performs the program management for the processing engines. System 10 also includes plain text interface processor (PTIP) 13 and cipher text interface processor (CTIP) 15 which provide external interfaces and signaling for system 10. The interface processors (IPs) also provide a high performance secure flexible buffer between an external host and the internal processing systems of system 10. System 10 also includes shared memory 18 which serves as an elastic buffer between KMCE 12 and PCP 17. System 10 also includes programmable interface 21 coupled with FILL and CIK ports 34. Testing of system 10

may be performed using test interface 20 which includes on-chip emulation and JTAG ports 35.

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw De

□ 21. Document ID: US 5949881 A

L2: Entry 21 of 28 File: USPT Sep 7, 1999

DOCUMENT-IDENTIFIER: US 5949881 A

TITLE: Apparatus and method for cryptographic companion imprinting

Detailed Description Text (16):

In block 240, this challenge is decrypted by the empowerment unit to determine the contents of the challenge. The contents of the challenge are then combined with another random bit string, which is newly generated for each challenge received, in order to form a resultant message (block 245). Thereafter, the resultant message is re-encrypted under the "shared secret" contained in memory accessible to the empowerment unit to create a response which is sent back to the cryptographic device (block 250). In block 255, the cryptographic device decrypts the response with the "shared secret" to verify that the response was processed by the empowerment unit. If the response was not produced by the empowerment unit but instead by another processor or by an external source, the laptop computer remains disabled (block 225); otherwise, the random bit string received from the empowerment unit is used as a temporary session key by the cryptographic processor to encrypt an "authorization message" and that message is sent to the source device's empowerment unit (block 250). In blocks 265-270, the empowerment unit decrypts the authorization message and compares this to a standard, predetermined authorization message (a message indicating that user verification and companion confirmation was successful). If these messages do not compare, the system processor as well as the laptop computer remains non-functional (block 225); otherwise, the empowerment unit enables the internal processing unit of the system processor and the laptop computer now becomes completely functional (blocks 275-280).

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw De

□ 22. Document ID: US 5428760 A

L2: Entry 22 of 28 File: USPT Jun 27, 1995

DOCUMENT-IDENTIFIER: US 5428760 A

TITLE: Circuitry and method for sharing internal microcontroller memory with an external processor

Brief Summary Text (14):

A system for sharing the internal memory space of a microcontroller with an external processor is described. The system includes a slave port, an interrupt server and a memory controller. The slave port serves as the external processor's window into microcontroller by passing address signals and data signals between the external processor and the internal memory space of the microcontroller. The slave port responds to a memory access request from the external processor by sending the interrupt server an interrupt signal. The interrupt server responds to the interrupt signal by determining the type of memory access and the physical address of the desired memory location. The interrupt server then generates appropriate memory control signals to cause the memory controller to execute the requested access. The memory controller reads and writes both the internal memory space of the microcontroller and the slave port, as appropriate, in response to memory control signals.

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw De

□ 23. Document ID: US 4953557 A

L2: Entry 23 of 28 File: USPT Sep 4, 1990

DOCUMENT-IDENTIFIER: US 4953557 A
TITLE: Blood pressure monitor

Brief Summary Text (12):
Through the invention makes sure that the pressure measurement produces correct values as long as the values determined by the two pressure sensors are similar, and as long as the supply and/or reference voltages are correct, and though correct supply and/or reference voltages are also a first condition for a correct operation of the electronic means included in said monitor, there is still the possibility that other components of the monitor are failing and are causing trouble by their failure. Thus, for example, trouble may be caused by a malfunction of the electronic means for calculating the blood pressure and for controlling the automated function of the monitor. In order to obviate severe consequences of such a malfunction, in a preferred embodiment of the invention said electronic means comprises a first processor for calculating the blood pressure and a second processor for controlling the internal and external data transfer. The failure sensing means comprises a shared memory coupled to said two processors and means for causing the deflation of said cuff when the data transfer via said shared memory fails. In this embodiment, the data transfer via said shared memory provides for a continuous check of the operation of said two processors, so that a malfunction of one of these processors also leads to deflation of the cuff and to switching off of the monitor.

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw De

□ 24. Document ID: US 4648065 A

L2: Entry 24 of 28 File: USPT Mar 3, 1987

DOCUMENT-IDENTIFIER: US 4648065 A
TITLE: Modified snapshot priority enabling two requestors to share a single memory port

Detailed Description Text (71):
Particularly in the preferred embodiment of the present invention, the two requestors which do share a single memory port are an external instruction processor, called IP3, and an internal maintenance processor, called MAINTENANCE EXERCISER, to the memory unit. It matters not that one such unit is internal, and one external, nor that one of such units is, particularly, involved primarily in the performance of a maintenance function. As will be seen from the following explanation, both such requestor units do make normal reading and writing requests of memory, and are both fully enabled for participation in priority through their shared single memory port. The circuit of the present invention is shown in FIG. 4, consisting of FIG. 4a and FIG. 4b. This circuit may be considered to be part of IP PRIORITY 68, previously seen in the block diagram of FIG. 1, although when shown therein only the memory requests of instruction processor 3 appeared as signal IP3 REQ on cable 203a while the request of memory from the internal maintenance exerciser was not shown within such high-level block diagram. The signal IP3 REQ part of cable of 203a shown in FIG. 1b is again shown as signal (H) IP3 REQUEST on line 203a (partial) in FIG. 4a. The request of the maintenance exerciser is shown as signal (H) EXER REQ on line 401 within FIG. 4a.

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw De

□ 25. Document ID: EP 1231543 A2

L2: Entry 25 of 28 File: EPAB Aug 14, 2002

PUB-NO: EP001231543A2
DOCUMENT-IDENTIFIER: EP 1231543 A2
TITLE: Shared-memory controller for use in a multimedia processor system

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw De

□ 26. Document ID: NN931237

L2: Entry 26 of 28 File: TDBD Dec 1, 1993

TDB-ACC-NO: NN931237

DISCLOSURE TITLE: Apparatus to Download and Verify Microcode onto Multiple Processors

PUBLICATION-DATA:
IBM Technical Disclosure Bulletin, December 1993, US

VOLUME NUMBER: 36
ISSUE NUMBER: 12
PAGE NUMBER: 37 - 38

SECURITY: Use, copying and distribution of this data is subject to the restictions in the Agreement For IBM TDB Database and Related Computer Databases. Unpublished - all rights reserved under the Copyright Laws of the United States. Contains confidential commercial information of IBM exempt from FOIA disclosure per 5 U.S.C. 552(b)(4) and protected under the Trade Secrets Act, 18 U.S.C. 1905.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

□ 27. Document ID: US 6668275 B1, WO 200144925 A2, AU 200129080 A, NO 200202853 A, EP 1238331 A2, KR 2002091057 A, JP 2003517677 W, NZ 520194 A

```
L2: Entry 27 of 28                File: DWPI                     Dec 23, 2003
```

```
DERWENT-ACC-NO: 2002-089523
DERWENT-WEEK: 200408
COPYRIGHT 2004 DERWENT INFORMATION LTD
```

TITLE: System for multiple-processor management using a core processor to interact with internal or external communication channels such as messaging queues, packet block transfers and shared memory schedules

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

□ 28. Document ID: US 5428760 A

```
L2: Entry 28 of 28                File: DWPI                     Jun 27, 1995
```

```
DERWENT-ACC-NO: 1995-240328
DERWENT-WEEK: 199531
COPYRIGHT 2004 DERWENT INFORMATION LTD
```

TITLE: Microcontroller for sharing internal memory with external processor - has slave port which receive logical address and control signal from processor and generates interrupt signal in response to control signal and uses interrupt server generates memory control signals in response to interrupt signal

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Terms	Documents
shar$3 adj3 (cache or memory or buffer) same internal near3 proces$4 same external near3 proces$4	28

Display Format: - Change Format

Previous Page Next Page Go to Doc#